

# BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

COMITE SCIENTIFIQUE :

N° 72 - DECEMBRE 2005

*Patrick Abellard*

*Françoise Adreit*

*Jalal Almhana*

*France Chappaz*

*M'hamed Charifi*

*Roger Cusin*

*Bernard Goossens*

*Patrick Isoardi*

*Robert Jacquier*

*Jean - Philippe Lehmann*

*Nadia Mesli*

*Patrick Sanchez*

*Rolland Stutzmann*

*André Tricot*

CORRESPONDANTS :

Afrique :

*Mohamed Tayeb Laskri*

Amériques :

*Sylvie Monjal*

Asie :

*Moussa HadjAli*

Europe :

*José Rouillard*

Océanie :

*Kalina Yacef*

**1 EDITORIAL**  
Utopie

*par Edmond Bianco*

**5 Les articles d'Edmond Bianco dans le bulletin**

*par Jean - Michel Knippel*

**9 Informatique**

*par Edmond Bianco*

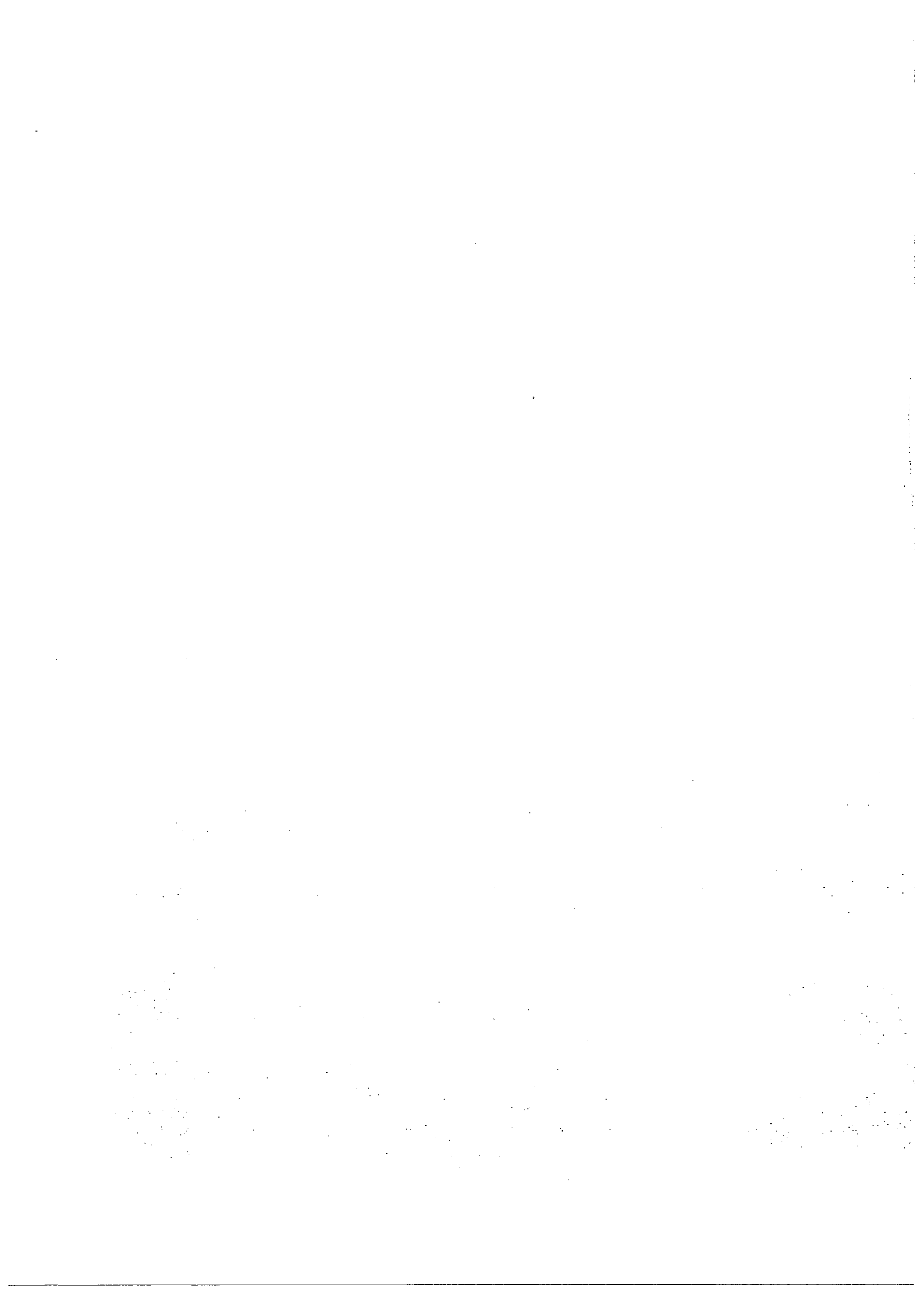
**23 VOZZAVEDIBISAR**

Travaux d'Edmond Bianco et dirigés par Edmond Bianco

*par Jean - Michel Knippel*

<http://scamup.univ-mrs.fr/biaa>

Publication trimestrielle, gratuite, de l'Université de Provence



# BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

N° 72- DECEMBRE 2005

DIRECTEUR :

*Jean - Michel Knippel*

REDACTEUR EN CHEF :

*Edmond Bianco*

REDACTEUR ADJOINT :

*Sami Hilala*

SECRETARIAT :

*Kalassoumi Adjilani*

Université de Provence  
Equipe Hermès. Case 33  
3, place Victor Hugo  
F - 13331 Marseille Cedex 3  
Téléphone: (0)4 91 10 62 30  
Télécopie : (0)4 91 50 91 10

**1 EDITORIAL**  
Utopie

*par Edmond Bianco*

DEPOSITAIRE :

Université de Provence  
Bibliothèque Vniversitaire  
1, place Victor Hugo  
F - 13331 Marseille Cedex 3  
Téléphone: (0)4 91 10 85 29  
Télécopie : (0)4 91 95 75 57

**5 Les articles d'Edmond Bianco dans le bulletin**

*par Jean - Michel Knippel*

IMPRIMEUR :

Université de Provence  
Service Reprographie  
3, place Victor Hugo  
F - 13331 Marseille Cedex 3  
Téléphone: (0)4 91 10 60 48

**9 Informatique**

*par Edmond Bianco*

**23 VOZZA VEDIBISAR**  
Travaux d'Edmond Bianco et dirigés par Edmond Bianco

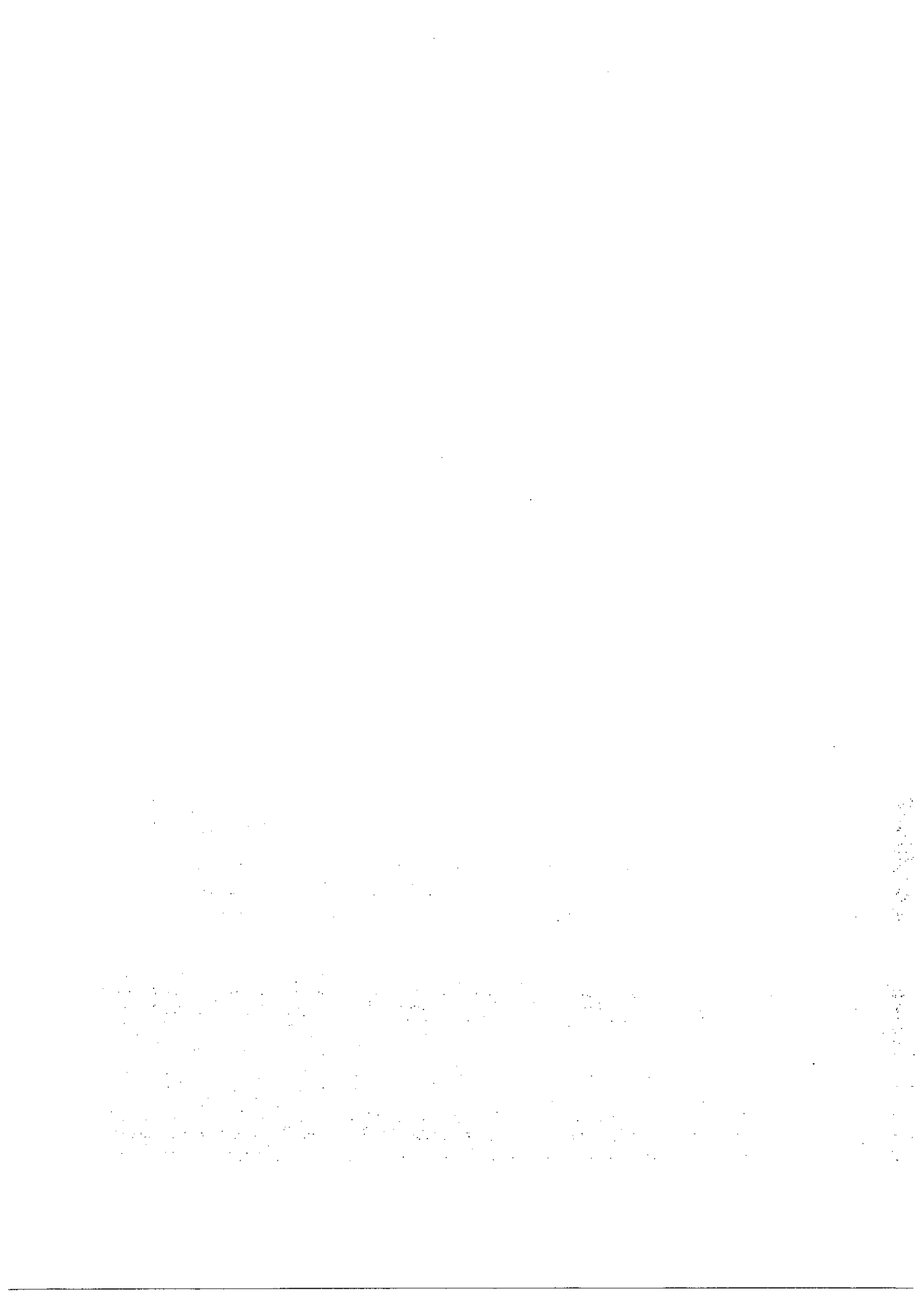
*par Jean - Michel Knippel*

<http://scamup.univ-mrs.fr/biaa>

Publication trimestrielle, gratuite, de l'Université de Provence

Impression : avril 2006

ISSN 0291 - 5413



## ÉDITORIAL

Utopie

*Edmond Bianco*

Notre société fonctionne sur un principe de base envahissant. Un mot est censé représenter cet étrange principe. Ce mot, parfaitement anodin, s'applique à quantité d'organisations, d'opérations, dont certaines sont proches de l'individu et d'autres, tellement éloignées qu'on a peine à les imaginer si l'on n'est pas spécialiste, partie prenante ou abusivement informé. L'économie. Ce mot recouvre à la fois la difficulté que rencontrent les ménages modestes pour survivre avec des salaires de misère dans un univers où ils ne maîtrisent rien, mais aussi le bricolage du budget de l'état, la spéculation financière internationale, le blanchiment de l'argent sale, l'organisation des paradis fiscaux, le financement des guerres.

Pendant longtemps cela représentait mettre de temps en temps quelques sous dans un bas de laine. Maintenant cela englobe nombre de théories fumeuses et parfaitement inadaptées dans la mesure où elles sont parfaitement incapables de prévoir quoi que ce soit.

### **Notion de travail.**

Dans des temps reculés nos ancêtres ne pouvaient évidemment survivre que dans les zones chaudes de la planète. L'envahissement des contrées à climats difficiles s'est très vraisemblablement réalisé sous la poussée d'une curiosité naturelle jointe à la nécessité de trouver de nouvelles ressources. Passer de la simple cueillette à la culture puis à la chasse, exigeait la fabrication d'un outillage adapté, pouvoir gratter la terre, passer de la « grotte-abri » à la cahute, passer du lancer de pierre au lancer de flèche. Le travail est apparu quand il a fallu tailler des outils, d'abord en pierres dures, puis des pointes de flèches pour les armes, puis se donner les moyens de coudre des vêtements, construire des abris. Les techniques s'affinant il y eut des spécialisations, il est apparu des corps de métiers, la technique a certainement aidé au développement de la démographie, en permettant l'envahissement de contrées toujours plus difficiles d'accès et en assurant une meilleure sécurité des populations. Mais d'autres problèmes ont évidemment surgi, liés à l'organisation de ces sociétés en train de s'agglomérer. Le rassemblement en villages, d'abord avec ses artisans, fondeurs, charrons, boulangers, tisserands etc. Certains métiers ne pouvant s'établir n'importe où, par exemple le fondeur ne peut pas travailler loin de ses approvisionnements en matériaux, les échanges sont devenus indispensables, d'où émergence du commerce par le troc.

Le village devenant cité, l'artisanat devint industrie ce qui provoqua l'apparition d'autres problèmes d'ordres sociaux. Le rapport qui se crée entre le paysan et son charron, qu'il va voir pour se faire faire une charrette ou une charrue, n'est pas le même que, désirant se moderniser, celui qu'il va établir avec le concessionnaire des tracteurs et matériels agricole. Le paysan et son charron sont liés par leur vie en commun dans le village, ils dépendent l'un de l'autre. Le concessionnaire des tracteurs lui, ne dépend que du groupe qui l'emploie. La vente du tracteur n'est pas liée au besoin direct de la machine, mais à l'application d'une machine, la publicité, qui devra persuader le paysan qu'il a absolument besoin de ce tracteur.

Au cours des siècles, l'évolution tribu, village, cité, n'a pas eu un développement linéaire, et tout s'est réalisé par essais successifs ayant donné lieu à des expériences plus ou moins longues, plus ou moins heureuses, mais variées dans le temps et dans l'espace.

**Remarque.** *Nous sommes en décembre, et, curieusement, malgré le mauvais temps et l'approche des fêtes, le ciel demeure zébré par le tonnerre des réacteurs de l'Armée de l'Air. Visiblement on brûle un maximum de carburant. Aucune menace extérieure ne justifie ce surentraînement de nos glorieux chasseurs. Mais. Mais en fin d'année il faut prouver qu'on a dépensé toutes les allocations énergétiques afin de ne pas déclencher l'horreur : la restriction des attributions pour l'année suivante. Il en est de même dans la plupart des administrations, un immense gaspillage, diraient des esprits non subtils qui n'ont pu saisir toute la finesse de cette logique administrative. Toujours ces mêmes esprits chagrins suggèrent que puisque des centaines de milliers de foyers bien trop démunis pour se payer du mazout, crèvent de froid pendant l'hiver, on pourrait les approvisionner gratuitement sans nuire gravement à notre défense nationale. Pour cela, poursuivent nos esprits contestataires, il faudrait, pour prendre une telle décision, que nos ministres aient pu posséder sur leurs épaules quelque chose qui ressemblât à une tête, avec dedans une quantité minimum de neurones. Ce ne semble pas être le cas.*

De puissantes cités se sont organisées, ont vécu comme des êtres vivants, ont grandi, se sont enrichies, puis ont décliné, quand elles n'ont pas été détruites par des hordes barbares qui se sont à leur tour civilisées au contact des vaincus et ont relancé une nouvelle civilisation. Récemment, l'empire Romain, suivi de l'empire de Charlemagne, ont subi ce trajet. L'époque moderne a subi la prédominance de l'Europe puissamment colonialiste. Et ce sont actuellement les barbares d'Amérique du Nord, les États-Unis, qui cherchent à éliminer les anciennes civilisations réparties dans le monde par leur « way of life » basé sur la mainmise du pétrole et de l'énergie.

La « Révolution Industrielle » du dix-neuvième siècle a définitivement coupé le monde en deux classes : les esclaves, surexploités et soumis à toutes les contraintes, et les maîtres qui disposent de l'outil de production des richesses, tandis que les banques stockent ces richesses et servent de régulateur du fonctionnement de la machine. Pour cela une petite notion de droit verrouillée par la caste des « Officiers ministériels », dresse une barrière infranchissable quand au partage des richesses.

Le vingtième siècle va apporter une nuance supplémentaire à ce tableau. Les luttes syndicales, sauvagement réprimées par la police qui défend « la propriété » et brisées par les guerres (Fortuites ?) ayant tout de même réussi à arracher quelques lois de protection sociales, ont permis aux esclaves de voir leur condition matérielle s'améliorer. Bien entendu dès que la droite, le parti des maîtres, revient au pouvoir, elle n'a de cesse que de détruire ces lois sociales, les retraites, la sécurité sociale, les protections du chômage, les protections des intermittents du spectacle, etc... Les esclaves étant censés disposer d'un certain niveau de vie, il ne s'agit pas de laisser gaspiller une telle richesse. Aussi la population se trouve-t-elle bombardée par une industrie lourde de l'abrutissement dont le but est de la persuader de « consommer ». (Cf. la déclaration de Lelay de TF1, novembre 2004). C'est un ensemble de notions encore plus subtil que la propriété, qui rend les choses si efficaces. Il s'agit de la « compétition » et de ses corollaires, la rentabilité.

**Remarque.** *Cet état de fait est extrêmement intéressant pour les banques. En effet quand on est pauvre et qu'on a vraiment envie d'essayer de vivre « comme tout le monde », c'est-à-dire de posséder son petit chez soi, il faut absolument passer par une banque pour se faire avancer l'argent. Une maison, si petite soit'elle coûte, cher. La banque, avant de lâcher ses sous, s'entoure de mille précautions, pour être sûre dans le pire des cas de récupérer plusieurs fois la mise, commerce oblige. Mais, même dans des cas parfaitement solvables, et c'est de moins en moins rare, chômage, maladie, divorce, la famille ne peut plus payer. La banque récupère alors la maison sans bien entendu rendre l'argent versé.*

La compétition se révèle le véritable cancer de notre société du vingt et unième siècle. Le « sport », il faut à tout prix vaincre, malheureusement la « machine humaine » n'est pas indéfiniment extensible, alors on y remédie par la dope. L'hypocrisie du système y montre alors toute sa force : on fait semblant de poursuivre quelques malheureux qui se sont fait épingler alors qu'on sait pertinemment que tous les champions se dopent. L'industrie pharmaceutique se fait un plaisir d'inventer de nouveaux produits qui permettent de gagner un peu de temps sur les contrôles.

## Évolution de la notion de travail.

Comme dans tous les domaines il y a ce qui se passe dans la vie courante et puis il y a des moyens linguistiques pour essayer de décrire les phénomènes. Là tout se complique déjà naturellement car chacun essaie de décrire de la manière qui réarrange le mieux. pour le patron, la grève est une catastrophe, pour l'employeur c'est un moyen désespéré de ne pas crever de faim. Se pencher sur la notion de travail c'est essayer de débrouiller une abstraction qui a été malmenée, embrouillée, par la littérature, le journalisme, la politique, les religions et tous modes de pensée tendancieux. On peut essayer, en tous cas de suivre le chemin parcouru par les occupations humaines, et essayer de constater leur évolution en fonction de la complexification de la société, et de suivre l'évolution de leur utilité et des nouveaux rôles qu'elles étaient appelées

à jouer. Le commerce, lui-même a puissamment évolué. On ne vend plus un produit, on vend l'image d'un produit, et pour renforcer cette image on en accole une autre, encore plus attractive, une fille sexy, par exemple. Maaaais le plus important n'est pas là. De plus en plus la production devient jetable. De telle manière que nos belles campagnes se chargent d'impressionnantes montagnes d'ordures, et que nous fonçons vers la « civilisation » du gaspillage intégral. Une « micropartie » de la société s'enrichit à mort en réduisant les frais à fond, pendant que l'immense majorité s'appauvrit au maximum car elle paye tous les déchets. Et quand ce phénomène n'est pas assez rapide aux yeux des « génies » qui mènent le monde, alors on organise une bonne guerre qui va détruire le maximum de richesses, lesquelles seront reconstruites par les vainqueurs de la guerre. Directement du producteur au consommateur et retour.

L'économie bien comprise c'est exactement cela !

Terminé, le 20 janvier 2005



## Les articles d'Edmond Bianco dans le bulletin

*par Jean - Michel Knippel*

Pendant une vingtaine d'années, Edmond Bianco a publié une série d'articles originaux sur ses travaux en informatique fondamentale. Seules deux années ont échappé à la règle : 1994 et 1995, années où il a pris sa retraite dans le Gard, mais le travail a continué dans l'optique de publier un livre. Tâche que certains membres de notre équipe scientifique continue.

Tous ses articles, il les a signés seul. Cela ne signifie pas un travail en solitaire, les travaux qu'il a dirigés et encadrés montrent la collaboration active avec un grand nombre de personnes. Nous en reparlerons dans les pages qui suivent l'article "Informatique".

Je ne résiste pas à l'envie de rappeler les noms de quelques amis avec lesquels il a travaillé depuis les années 1960 et sa montée à Paris, pour rejoindre le laboratoire de calcul numérique du CNRS. "Des caves de l'institut Henri Poincaré à la terre promise de la rue du Maroc", il a participé à ALGOL 60 avec Lemaire. Je citerai encore ceux qui donneront matière à des travaux : Louis Nolin, issu de la logique ; André Lentin, rompu à l'algèbre ; Maurice Gross, du fameux "Gross et Lentin" sur les grammaires formelles. La liste est encore longue, et j'en oublie : Arzac, Dupuy, Nivat, de Possel, Schutzenberger. Tout ce beau monde se connut à Paris. Ensuite, à Strasbourg, il travailla avec Flajolet, Isoardi, Jammes, Julien, Lehmann, Robinet, Stutzmann. De retour à Marseille, dans les années 1970, il y rencontre d'autres universitaires : Berstel, Bresson, Colmerauer, Cusin, Jacquier, Le Moigne, Livet, Meloni, Risset.

Plus qu'un long discours sur ces travaux, nous avons choisi de republier l'article du bulletin N° 20 de juin 1988 "Informatique", il présente avec clareté "ce qui constitue le noyau de l'informatique fondamentale". C'est un article à "caractère unique, très pédagogique et même, d'une certaine façon, divertissant.". Je reprends ici les propos de Jean-Luc Gaudiot, professeur à l'université de Californie Irvine (UCI), de la préface du livre de notre ami Bernard Goossens "Architecture et micro-architecture des processeurs", paru en 2002 chez Springer.

Le processeur expérimental MCA-0  
Bulletin N° 0. pp. 6 à 36. Mars 1981

Le processeur expérimental MCA-0 : suite  
Bulletin N° 1. pp. 6 à 11. Juin 1981

Machines, langages, compilation  
Bulletin N° 2. pp. 6 à 36. Mars 1982

Autojection et compilateurs  
Bulletin N° 2. pp. 50 à 56. Mars 1982

Autojection et compilateurs : "suite"  
Bulletin N° 3. pp. 62 à 77. Juin 1982

Autojection et compilateurs : "suite de suite"  
Bulletin N° 4. pp. 23 à 43. Décembre 1982

Autojections et compilateurs. Procédure formelle symbolique  
Bulletin N° 6. pp. 50 à 65. Décembre 1983

Autojection et systèmes. Procédure formelle symbolique  
Bulletin N° 8. pp. 41 à 48. Juin 1984

Notion de système : système logiciel et système machine  
Bulletin N° 9. pp. 8 à 30. Décembre 1984

Notion de système : système logiciel et système machine : "suite"  
Bulletin N° 10. pp. 4 à 33. Mars 1985

Un théorème important de l'informatique : théorie de l'index  
Bulletin N° 11. pp. 25 à 36. Juin 1985

Notion de système : système logiciel et système machine : "suite de suite"  
Bulletin N° 12. pp. 26 à 45. Décembre 1985

Autre théorème important de l'informatique :  
puissance des machines à files multiples  
Bulletin N° 13. pp. 6 à 33. Mars 1986

Notion d'insertion dans la machine de Nolin  
Bulletin N° 15. pp. 29 à 42. Décembre 1986

Machines formelles et échanges. Machines de distribution  
Bulletin N° 17. pp. 5 à 20. Juin 1987

La machine plate  
Bulletin N° 18. pp. 5 à 31. Décembre 1987

Notion d'autojectivité. Notion de statut  
Bulletin N° 19. pp. 5 à 21. Mars 1988

**Informatique**  
**Bulletin N° 20. pp. 5 à 24. Juin 1988**

Système universel  
Bulletin N° 21. pp. 27 à 37. Décembre 1988

Système universel : " suite "  
Bulletin N° 22. pp. 6 à 17. Mars 1989

Système universel : " suite de suite "  
Bulletin N° 23 pp. 6 à 25. Juin 1989

Système de Post : construction de phrases  
Bulletin N° 25. pp. 18 à 25. Mars 1990

La machine à instruction unique  
Bulletin N° 26. pp. 22 à 29. Juin 1990

Dynamique de l'informatique  
Bulletin N° 27. pp. 38 à 41. Décembre 1990

Statuts  
Bulletin N° 29. pp. 5 à 18. Juin 1991

Informatique, vérité, théorème de Gödel  
Bulletin N° 30. pp. 9 à 20. Décembre 1991

Le processeur de la procédure formelle  
Bulletin N° 31. pp. 7 à 27. Mars 1992

Le processeur de la procédure formelle : suite  
Bulletin N° 32. pp. 5 à 20. Juin 1992

Statuts de variables  
Bulletin N° 33. pp. 4 à 17. Décembre 1992

Insertion fractionnée  
Bulletin N° 34. pp. 11 à 28 . Mars 1993

Chapitre 1. Notions de calcul  
Bulletin N° 45. pp. 23 à 28. Décembre 1996

Chapitre 2. Une machine élémentaire  
Bulletin N° 47. (pp. 3 à 14. Juin 1997

Chapitre 3. Compléments à la machine de Nolin  
Bulletin N° 47. pp. 15 à 18. Juin 1997

Chapitre 4. La machine formelle  
Bulletin N° 47. pp. 19 à 26. Juin 1997

Chapitre 5. La machine universelle  
Bulletin N° 51. pp. 3 à 16. Décembre 1998

Chapitre 6. Notion de système  
Bulletin N° 51. pp. 17 à 34. Décembre 1998

Chapitre 7. Le processeur  
Bulletin N° 53. pp. 5 à 26. Juin 1999

Chapitre 8. La machine universelle de la procédure formelle  
Bulletin N° 56. pp. 7 à 26. Juin 2000

## Informatique

*Edmond Bianco*

Ce mot, à la fois adjectif et substantif est devenu d'un usage courant. Au point qu'à la moindre occasion pour peu qu'il s'agisse d'un vague calcul, on l'emploie. Il a même remplacé les menaces habituelles qu'on utilise à l'égard des enfants qui ne sont pas sages : si tu ne fais pas ce que je te dis, gare à l'informatique. Pourquoi s'insurger contre ce constat ? pourquoi vouloir se conduire comme un académicien ?

Le mot est bien utile, d'abord il fait sérieux dans la conversation et puis il comble un certain vide, et sa réussite sociale est tout-à-fait récente. Il y a peu encore, l'informaticien était cette sorte de praticien peu ragoûtant, considéré par les purs théoriciens comme un vil tâcheron. Et puis le fric aidant et les services rendus à l'administration ...

Il n'est pas de mathématicien désormais qui ne tapote en catimini sur son petit clavier, il n'est plus de production abstraite qui ne fasse allusion au nouveau dieu de la pensée pure, j'ai cité l'informatique. Il vaut mieux prendre le train déjà bien lancé que de le rater.

Peut-être serait-il possible de faire un peu le point de la situation pour voir comment, au moins du côté des spécialistes, se concrétise un peu l'informatique, celle qu'on rencontre forcément quand on veut calculer, en mettant un peu de côté l'informatique de salon, gymnastique des beaux esprits.

L'informatique apparaît comme bâtie sur deux pôles, au moins considérés comme tels : le câblé et le logiciel. De tous temps, c'est-à-dire depuis le début de l'ère Turing-Von Neumann, câblé et logiciel se sont développés quasi indépendamment, au sein d'équipes qui s'ignoraient superbement. Peut-on toutefois estimer qu'il y a véritable disjonction entre ces deux domaines, cela ne me paraît pas un véritable problème et de toute manière les techniques les plus modernes permettent de montrer facilement comment ils peuvent s'interpénétrer. Et depuis peu, il est devenu très à la mode de se préoccuper des deux à la fois.

**Je vais essayer de faire une approche des problèmes centraux de l'informatique. D'abord avec la pratique, on s'aperçoit que quelle que soit la difficulté rencontrée, il va falloir l'aborder sous deux angles différents. Une première approche du calcul se déroule en fini-borné et une deuxième déborde le plus souvent dans le fini-illimité. Ceci est tout simplement dû à une raison pratique, toute matérielle.**

Côté configurations on ne sait pas construire des mémoires automatiquement extensibles, par contre on sait comment mettre des rallonges à volonté. Côté algorithmes, un processus qui serait indéfini dans le temps serait inutilisable, alors on le découpe en éléments parfaitement définis, dont la répétition peut, elle, devenir aussi longue que nécessaire.

Cette simple contrainte a des conséquences pratiques difficiles à éviter. Ainsi une mémoire d'ordinateur est construite en deux parties, la mémoire centrale et la mémoire de masse, externe. On profite de ce fait pour différencier au maximum les qualités requises pour l'un et l'autre organe. La mémoire centrale où s'effectuent tous les calculs sera d'accès le plus rapide possible et pour atteindre facilement chaque case on invente l'index dont le contenu est considéré comme un nom de case. Et dont l'emploi est précisément justifié par la dimension bornée de la mémoire. La mémoire externe doit être volumineuse au maximum, même au détriment de la vitesse et de la facilité d'accès. C'est ainsi que sur un même dérouleur on peut faire défiler successivement autant de rubans magnétiques, de disques ou disquettes qu'on le désire. Et j'oublie volontiers cartes et rubans perforés.

Parallèlement à cela se posent quelques questions sur le domaine du calcul lui-même. En construisant des machines dotées de proportions différentes ne risque-t-on pas de créer artificiellement des domaines de calcul qui ne se recouvrent pas exactement ? Un calcul qui pourrait se faire sur une machine mais qui ne le pourrait pas sur telle autre ?

D'où l'intérêt fondamental de vérifier que tout ce qui peut se faire sur une machine peut également se faire sur l'autre. Quand on a ainsi fait le tour de toutes les machines connues, au moins peut-on être un peu rassuré.

Ces propriétés sont mises en évidence par une classe de théorèmes situés véritablement à la base de l'informatique et qui définissent la puissance réelle de l'outil. Les techniques de démonstration employées, qui mènent toujours à des propriétés d'existence, voire de non existence, ont en général des aspects

très constructifs. On simule une machine sur une autre, cela permet de mesurer la commodité des concepts utilisés.

On fait essentiellement ce travail sur deux machines voisines, l'une des deux possédant un mécanisme de plus dont on peut sonder l'efficacité. Dans une certaine mesure, si ce mécanisme se révèle à la fois utile et compliqué à simuler, on peut en déduire l'économie qu'il permet de réaliser.

**C'est ainsi qu'on s'aperçoit qu'un demi ruban est suffisant pour réaliser tous les calculs, par contre, un ruban illimité dans les deux sens facilite bien les choses. De la même manière un seul ruban est aussi efficace qu'un nombre quelconque de rubans. Mais un bon nombre de rubans peut servir à simplifier telle catégorie d'algorithmes.**

**C'est alors que le fait d'isoler un morceau de ruban limité à droite et à gauche ce qui permet l'accès direct à chacune des cases, bouleverse complètement l'art du calcul, au moins pour ce qui est du fini-borné. Néanmoins la propriété globale se conserve pour le fini-illimité, quand il s'agit de ranger sur mémoire de masse.**

Une machine quelconque s'appréhende selon deux aspects qui dépendent fortement l'un de l'autre. La machine est un mécanisme, mais on la fait fonctionner avec un langage. Le langage tient compte de la structure du mécanisme, et le mécanisme est précisément conçu pour à la fois décoder le texte du langage codé, et agir sur l'information à traiter, également codée.

Ce mécanisme peut être décrit à son tour comme un simple programme, et là deux solutions se présentent. On peut utiliser soit un langage de programmation qu'on qualifiera de symbolique, soit un langage qui décrit directement un câblage. C'est par là que se rejoignent précisément logiciel et câblage. Pour passer de la description purement symbolique à la description, je dirai pour l'instant électronique, on utilise tout un spectre de langages dont le dernier est l'image même du câblage. Voilà un chapitre qui montre que ces deux domaines sont loin d'être indépendants.

Le premier contact utilisateur de l'informatique se fait par le langage. C'est la raison essentielle pour laquelle, encore, informatique et programmation demeurent plus ou moins synonyme pour le public. Vouloir se servir d'une machine c'est avant tout apprendre à écrire au moyen d'un langage très synthétique. Au commencement il n'existait que des langages machine mais la

difficulté de leur emploi en réduisait largement la diffusion, aussi pour pallier à cet inconvénient on s'est mis à développer rapidement les langages qui se sont enrichis et se sont trouvés classés en deux catégories, l'une qui comporte les langages machines, destinés à travailler directement sur la machine et d'usage difficile, l'autre qui comporte les langages symboliques, plus proches des habitudes culturelles de l'utilisateur et d'usage plus agréable.

Ces langages symboliques se sont alors développés à profusion avec un succès d'autant plus débordant qu'ils permettaient l'accès de l'ordinateur au plus grand nombre.

Formellement tous ces langages ont des structures relativement simples et systématiques dont l'étude faisait l'objet de la combinatoire et de la théorie des langages. Pour certains, et pendant longtemps, c'était cela l'informatique. On ne savait encore que peu de choses sur la sémantique dans la mesure où tous les travaux importants de ce domaine n'avaient pu être encore rassemblés pour constituer un tout. Et puis surtout c'était un domaine nouveau de la réflexion dont on pouvait extraire nombre de petites vérités rigoureuses, tradition oblige. L'intérêt des langages pour cette informatique qui naissait est indéniable, il s'imposait d'attribuer à la jeune informatique ses lettres de noblesse. Et puis cela créait un débouché sur des carrières par ailleurs bien bouchées. Les choses ont largement évolué et l'étude de la sémantique me paraît avoir normalement accaparé le cœur de l'informatique aussi je suis tenté de penser que la théorie formelle des langages, domaine de la connaissance relativement fermé, se place plutôt en "épi-informatique", dans la mesure où ce domaine bien qu'indispensable à la formalisation de la donnée informatique, ne saurait apporter grand chose à la résolution des problèmes de sémantique qui constituent le développement actuel de l'informatique.

Très généralement dans les langages, et quels que soient ces langages, se présentent des propriétés que j'aurais tendance à qualifier de fondamentales, car elles dépendent non de la spécificité de forme du langage mais de lois sémantiques nées des usages.

Ainsi, c'est par la pratique qu'on s'est aperçu qu'il était bon de construire les programmes par petits morceaux en essayant de condenser dans chacun, des éléments de calcul les plus universels possible pour économiser le travail au maximum. Il fallait alors organiser ces programmes pour les rendre utilisables facilement quel que soit l'environnement. Notion de procédure. A partir de là, on définit une opération d'insertion de procédure et de retour systématique dont



la sémantique est parfaitement indépendante du calcul envisagé.

**Le principe de Von Neumann permet de concevoir une machine qui ne serait pas spécialisée dans un calcul déterminé et immuable, mais capable de lire un code dans sa mémoire, qu'elle interprète comme un programme en modifiant par ailleurs une configuration à calculer. L'algorithme de cette machine capable d'interpréter un code, et dont on montre l'existence, est une machine universelle. La notion de machine universelle est encore indépendante du calcul que l'on veut réaliser.**

Cette notion est, en plus, très constructive car elle permet de mettre assez facilement en œuvre des instructions qui n'existent pas dans le langage de la machine support. C'est par ce moyen qu'on peut introduire aisément l'insertion de procédure, par exemple.

Le programme apparaît alors sous sa forme définitive, une sorte de code qu'il devient possible d'introduire en mémoire comme une simple donnée. Mais, qu'il vaut mieux éviter de confondre avec une simple donnée, problème d'organisation du calcul. La condensation de la sémantique dans les codes ayant signification de programme est un problème important de l'informatique.

Les langages symboliques de programmation ne prennent vraiment tout leur intérêt que dans la mesure où on peut les traduire en langage machine afin de les rendre utilisables. Ceci se réalise au moyen de la notion de compilateur. La caractéristique essentielle de ces langages par rapport aux langages machines tient dans la représentation des objets de la programmation par des symboles, dont la commodité réside précisément en ce qu'ils sont indépendants de toute géométrie de mémoire. Car la variable à forme symbolique est bien plus proche de nos coutumes culturelles de pensée.

On en profite également pour représenter sous forme simple, diverses fonctions plus ou moins complexes. A charge pour le compilateur d'établir une correspondance entre toute phrase possible du langage et un programme "dérivable" sur la machine support. Bien entendu, l'équivalence sémantique stricte doit être respectée entre les deux constructions. Dans la réalité, cela n'est pas toujours rigoureusement vrai. Car trop souvent la représentation des objets symboliques laisse à désirer.

La compilation est encore une notion, née de la nécessité à la fois d'utiliser des langages de programmation évolués pour rendre l'ordinateur accessible au non

spécialiste, et de rendre effectivement calculables les programmes ainsi obtenus. Donc notion parfaitement indépendante du problème que se pose l'utilisateur, mais en permanence sous-jacente car sans cela l'ordinateur demeure quasi inabordable.

Quand on poursuit cette quête qui consiste à faire de cet organe un instrument de plus en plus agréable d'accès, de plus en plus "convivial", force est de constater que le compilateur est en lui-même un instrument incomplet. En effet.

Le programme traité présente deux aspects qui vont encore nous obliger à réfléchir. Le compilateur transforme la phrase en une suite de codes interprétables qui sont construits pour contenir la sémantique à la fois nécessaire et suffisante. Le code sera repris ultérieurement par un compilateur dérouleur, forme matérialisée de la machine universelle, afin d'obtenir le calcul désiré. Quelquefois on alterne l'action du traducteur et celle du dérouleur afin d'obtenir un calcul en "interprété".

De toute manière, après traduction il existe en mémoire un nouveau corps qui se trouve être l'image du programme symbolique. Et il faut compter avec lui. S'il doit servir plusieurs fois, ce qui est le cas général, on va devoir le manipuler en bloc ne serait-ce que dans un but de sauvegarde en mémoire de masse pour ne pas encombrer la mémoire centrale quand ce n'est pas nécessaire. Puis il faudra le rappeler en temps opportun. Voilà donc un traitement qui ne peut faire raisonnablement partie de ce qu'il est possible d'attendre d'un compilateur, tout comme la constitution des configurations de travail qu'il faut bien soumettre au déroulement du programme. Plus généralement, ainsi que toute organisation des moyens disponibles à partager entre plusieurs compilateurs.

**Il apparaît ainsi une autre catégorie de traitement dont le domaine d'application se situe à l'articulation entre le fini-borné, matérialisé par la mémoire centrale, et le fini-illimité matérialisé, lui par la mémoire de masse. La notion de système informatique surgit au moment où l'on veut rendre utilisable les produits de la compilation. Les choses se compliquent rapidement car on constate qu'un programme quelconque est a priori du domaine du fini-illimité. Le compilateur ne peut apprécier le volume exigé par la phrase qu'il traduit qu'à l'instant où elle s'achève.**

Entre temps il aura fallu gérer la place. Il n'est pas raisonnable de confier cette tâche au compilateur, qui, par définition dépend des propriétés du langage symbolique.

De semblables remarques se font également à propos des structures de l'algorithme. On peut observer par exemple quand on part de la notion de procédure, qu'un programme est constitué d'un jeu de procédures qui s'insèrent les unes dans les autres. Le programme lui-même est une procédure. La question se pose alors de savoir, où lui s'insère. Je peux dire qu'il s'insère dans le système, mais s'il n'existe qu'une seule structure possible, la procédure, on est tenté de se poser la question dangereuse qui risque de déboucher sur un certain mysticisme : où s'insère le système. **Il vaut mieux définir le système comme une procédure à la fois spéciale et spécifique.**

**On constate ainsi que les notions de compilation et de système, quoique conceptuellement bien distinctes vont devoir impitoyablement se mélanger.**

Là encore la notion de système est totalement indépendante de la tâche à dérouler.

Si l'on n'y prend pas garde ou si l'on réalise des études locales, on court le risque d'aboutir à des constructions effroyablement compliquées et peu fiables. Une autre étape d'évaluation sémantique s'ouvre ainsi à la conception de systèmes et de jeux de compilateurs cohérents.

**Par principe le système assure la gestion du fini illimité, c'est pour cette raison que le côté conversationnel entre naturellement dans son objectif, car il s'agit là d'un échange imprévisible et sans fin. Le système va se trouver chargé de la communication "conviviale" et du traitement ergonomique qui lui est lié.**

Là se trouve une véritable mine de recherches sur la structuration sémantique des compilateurs et des systèmes. En gros deux thèses s'affrontent basées, l'une sur un procédé technique l'autre sur un principe d'auto-organisation. Jusqu'à présent a prévalu une organisation à base d'interruptions dues à un mécanisme spécifique. Les compilateurs sont pratiquement quelconques, et c'est le système qui a la charge du découpage dans le temps, donc la charge également d'une sauvegarde et d'une gestion de l'information mise en péril par une interruption aveugle.

Un autre principe, plus moderne, se fonde sur l'autojectivité ou auto structuration des compilateurs. Le découpage dans le temps devient une propriété de structure du compilateur lui-même. Cela implique que d'autres méthodes soient mises en œuvre pour la construction des compilateurs, du genre des compilateurs compacts.

Cette deuxième méthode permet essentiellement de décrire entièrement les propriétés de la gestion complète de l'ordinateur, donc de mettre en évidence les points de condensation des notions importantes. Par exemple l'insertion fractionnée qui joue le rôle de transmetteur d'information entre système et compilateurs.

La pratique montre qu'il existe des sortes différents de systèmes suivant les proportions entre les diverses tâches qui lui sont dévolues. Ainsi la classe des systèmes chargés de répartir le travail entre divers utilisateurs qualifiés de scientifiques est assez différente de la classe des systèmes chargés d'organiser du contrôle de processus. Là encore est ouvert le domaines de l'étude de la structuration sémantique de ces gros ensembles.

Parallèlement à ce genre de recherches, sont menées des études sur la construction de langages très spécialisés destinés précisément à décrire et matérialiser les théories que l'on peut échafauder sur les problèmes fondamentaux. C'est ainsi qu'on s'est rendu compte que certains langages symboliques permettaient de construire très facilement toute la partie fonctionnelle des compilateurs, mais très difficilement tout ce qui permet au compilateur d'être rattaché à son environnement. Alors on en est revenu à des expériences du type langage C qu'on a maintenu proche des langages machines. De la sorte, on perdait un peu sur le premier tableau mais on regagnait sur le second. Ce langage répondait à la fois à un besoin et en même temps n'était adapté à aucun objectif précis. La nécessité obligeait rapidement à le bricoler, le faisant ainsi considérablement vieillir.

Mais la mode aidant, il était lancé, il était lancé.

**Ce rapide survol de la situation au temps présent me paraît suffisant pour raisonnablement cerner ce qui constitue le noyau de l'informatique fondamentale. C'est en fait la théorie de l'outil de travail, indépendamment des tâches qui lui sont assignées, mais en tenant compte de l'obligation de plus en plus sévère d'offrir un service présentable au plus grand nombre.**

De telle sorte que le sens de l'informatique c'est organiser la "complexification" de l'outil afin d'améliorer et simplifier au maximum le service rendu, tout en conservant une fiabilité acceptable.

Quelle peut être alors l'ouverture que l'on peut imaginer pour cette discipline ? On se heurte en ce point à une contradiction tout à fait fondamentale. L'outil qu'on utilise couramment, qui s'est largement diffusé, est puissant et rapide mais il est rédhibitoirement stupide.

Or, l'ordinateur était à peine construit qu'on s'est demandé si le mythe de Frankenstein n'était pas désormais à la portée de la main. Nombre d'expériences de qualité furent tentées, et dans une certaine mesure réussies. On prouvait qu'il était possible de mettre au point des programmes très complexes capables de simuler certains processus bien délimités de la pensée. Mais hélas toute étincelle d'intelligence en était encore exclue, ces programmes demeuraient incompetents pour l'extérieur des domaines rigoureusement bornés pour lesquels ils avaient été construits.

Doit-on en conclure pour autant qu'il faut déduire de ces expériences que jamais il n'existera parmi nous de docteur Frankenstein, l'état de nos connaissances ne permet pas d'en décider. Déjà, si les machines ne se sont, quelle que soit la programmation qu'on ait pu leur offrir, pas révélées plus intelligentes, les hommes, eux ont dû faire de gros efforts, et peut-être le résultat de ces efforts a-t-il accru quelque peu le potentiel d'intelligence de l'Humanité. A coup sûr de l'intelligence technique, mais il faut rajouter beaucoup d'optimisme pour prétendre de l'intelligence tout court.

L'enjeu n'est rien moins que de donner à un automate ou un ensemble d'automates un certain degré d'autonomie, donc d'individualité, donc lui fournir une motivation qui le pousse à comprendre pour pouvoir agir. Quand je constate dans quel milieu, notre époque, pourraient naître de tels monstres, je suis amené à me poser la question fondamentale.

Dans un univers du profit à tout prix et de son instrument aveugle la compétition, alors que pèsent si peu les populations par exemple dans la ruée à l'uranium, bref dans une civilisation qui sue le pouvoir pour le pouvoir par tous ses pores, peut-être faudrait-il attendre encore un peu. On peut espérer que toutes les "micro-féodalités" démocratiquement diffusées auront un jour usé leur pouvoir au coin de la vanité et qu'alors un apport d'un peu d'intelligence mécanique pourrait être utile à autre chose qu'à accroître le pouvoir des ganaches.

Bien entendu rien n'est jamais si simple qu'on puisse l'enfermer en quelques mots, et la réflexion sur les systèmes complexes permettra-t-elle peut-être de

faire progresser la sociabilité de l'être "civilisé". Il faut le souhaiter s'il est difficile de faire plus.

La rêverie utopique est parfaitement indispensable à la progression de la pensée mais il faut revenir de temps en temps à la simple technique prosaïque. J'ai l'intime conviction que la machine de Turing telle que nous la connaissons est insuffisante pour dépasser le stade du simple automate réflexe.

Peut-être peut-on faire l'hypothèse hardie que la construction d'un logiciel suffisamment complexe pourrait dépasser le point à partir duquel un organisme déjà suffisamment complexe peut parvenir de lui-même à "s'autocomplexifier." Mais ceci est largement insuffisant pour bien cerner la question. Il est assez facile de construire un automate capable d'accroître sa propre complexité, mais pour qu'on puisse parler d'intelligence, cet accroissement est loin d'être quelconque. Il faudrait qu'il soit capable en outre de trier parmi ses productions celles qui peuvent lui être utiles en tenant compte d'une foule de paramètres qui lui sont étrangers qu'il doit par là même pouvoir percevoir et assimiler.

En quelque sorte "l'autocomplexification" doit être telle qu'elle mène à une complexité dont la cohérence avec la complexité du monde extérieur ait tendance à s'améliorer.

Il paraît évident que l'un des inconvénients majeurs que présentent nos ordinateurs, je dirai en bref la machine de Turing, c'est essentiellement de disposer de langages dont la forme est totalement indépendante de la signification. J'entends par là que si je soumets une phrase d'un langage quelconque à un calcul de même nature que celui que permet le langage dont la phrase est issue, j'obtiens une nouvelle phrase, qui, si elle déroulée ne peut absolument rien donner de cohérent avec la phrase d'origine. De telle sorte qu'il n'est pas possible de définir une sorte de continuité ou de régularité dans le calcul de phrase.

La recherche actuelle est très polarisée sur les problèmes que soulève l'apprentissage automatique, ainsi d'ailleurs que sur les réseaux de simulation neuronique. On étudie également les algorithmes compliqués destinés à prendre en compte certains problèmes de la linguistique des langues naturelles. Mais ce faisant on n'arrête pas de se heurter aux bornes que présente la machine.

**J'ai donc tout lieu de penser que le prochain champ d'application de l'informatique fondamentale se construira autour de la conception de machines dont la puissance débordera, si peu que ce soit, du champ de l'effectivement calculable. De nouvelles conceptions de machines sans doute moins fiables que la machine de Turing, mais incomparablement plus puissantes dans la mesure où, plus souples, elles disposeront de ce qui leur permettra de percevoir, d'assimiler à l'aide d'une sorte de réflexion, de créer des images comparables aux images dues à leur perception, et par ces comparaisons de bâtir des hypothèses vérifiées ou infirmées par de nouvelles perceptions .**

En quelque sorte des machines ayant un fonctionnement interne permanent sur elles-mêmes, comme une sorte de réflexion, et qui consultent de temps à autre l'univers dans lequel elles baignent, pour enrichir leur pouvoir de perception et d'intervention sur cet univers.

C'est dans cette direction que la notion de système devrait pouvoir s'enrichir.

On peut ainsi résumer la situation en disant que le champ actuel de l'informatique fondamentale recouvre le domaine des propriétés communes au logiciel et au câblé qui sont induites par la notion de service. L'utilisateur aborde l'ordinateur avec son problème, qui lui est spécifique. La difficulté qui l'attend se partage entre la description de son algorithme, je dirai la difficulté inhérente, et l'adaptation de son algorithme à l'environnement que représente le système de l'ordinateur, la difficulté d'adaptation.

Autant il est naturel que l'utilisateur prenne en charge la difficulté inhérente, l'informatique doit offrir à sa préoccupation le moyen d'expression le plus approprié, premier apport, autant il est nécessaire que la difficulté d'adaptation lui soit réduite au maximum, et dans la mesure du possible présentée de façon qu'elle ne heurte pas sa culture, deuxième apport. Voilà pour les intentions.

La puissance apparente de l'ordinateur est devenue telle, surtout dans certaines utilisations collectives à but plus ou moins politique, et dans certaines utilisations fascinantes de type jeu, qu'il est indispensable de démythifier et même de démystifier en permanence son rôle. Il est devenu l'instrument le plus puissant de pouvoir centralisateur. On le présente à la fois comme une sorte d'arbitre impartial et comme une pythie dont on aimerait bien que les prévisions deviennent réalité. Acte de persuasion dont on peut rêver qu'il est efficace. Arcanes et foi.

A la charge de l'informatique fondamentale vont se trouver toutes les propriétés de l'outil étudié du point de vue câblé et du point de vue logiciel. Essentiellement les règles qui relient le domaine du câblé au domaine du logiciel. Comment reflue la sémantique de l'un à l'autre quand on apporte des modifications à l'un. De là découlent l'explication des choix de structures de l'ordinateur, et les conséquences à en tirer pour son utilisation.

A partir d'un choix de structure, se trouve disponible le langage de base, support du logiciel. C'est un univers fluide, difficile à appréhender, dans lequel l'expérience, la pratique, vont permettre de poser des jalons et monter des sortes de casiers qui vont faciliter l'occupation du terrain. Si l'on y réfléchit bien, la notion de procédure dépasse largement le domaine de la pensée informatique.

La mise en œuvre de la notion de computation, de calcul effectivement réalisé, oblige à repenser le rapport déjà établi entre câblé et logiciel. La notion de machine simple est la matérialisation d'un algorithme, sa réalisation pratique est une opération délicate et coûteuse. D'où la création de la notion de machine universelle qui est une machine simple mais qui bouleverse la notion de matérialisation de l'algorithme en en faisant une donnée traitée.

La dualité d'aspect du calcul, constituée normalement d'un point de vue culturel qui nous est propre par le doublet : donnée à traiter, algorithme de traitement, nous permet alors de transcender définitivement la notion de programme\*. On trouve dès lors côte à côte dans la mémoire : les données et un code qui représente l'algorithme.

La rusticité des langages machines liée bien évidemment à la difficulté de câblage de machines universelles trop complexes a poussé fortement à l'introduction de la compilation. Mais d'autres arguments peuvent être également avancés. Par exemple, si l'on considère que les langages symboliques sont d'utilisation spécifique, c'est là une part importante de leur propriété, ils sont optimisés pour une classe de problèmes, imposer un tel langage à un ordinateur revient à en limiter le domaine d'applications. Les langages machines, s'ils sont rustiques et pénibles à utiliser, possèdent par contre, la plus grande souplesse d'emploi.

*\* Je n'ai pu résister au plaisir d'employer un peu de langue de bois, mais il aurait été plus simple de dire :*

*"On a l'habitude de concevoir un calcul comme un jeu de règles qu'on applique sur un jeu de données, l'algorithme y apparaît alors sous une forme nouvelle."*



Enfin tout en haut de l'édifice, se place le système destiné à gérer convenablement l'ensemble et à assurer la présentation la plus ergonomique possible.

Cette fonction système exécutée entièrement à la main dans les premiers âges de l'ordinateur est de nos jours en grande partie automatisée.

**Voilà donc un bref tour d'horizon qu'il me paraissait indispensable de faire. l'expression d'une autre notion est d'usage courant c'est celle d'informatique théorique. Elle ne me semble avoir de sens que par opposition à une informatique pratique, dont hélas personne n'ose se réclamer. Ajoutons à cela qu'il n'est rien moins qu'évident de tracer une démarcation nette entre ce qui est pratique et ce qui est théorique en informatique.**

Toutefois, je pense que cela n'enlève rien à l'intérêt de ce concept d'informatique théorique dont l'utilité m'apparaît plutôt d'ordre administratif. C'est alors un fourre-tout dans lequel même les domaines "épi-informatiques" lointains peuvent prendre place, simple question de survie. Mais acte méritoire tout de même quand on songe à la reconversion.

**Ce propos n'est pas destiné à définir une fois pour toute dans un esprit partisan la notion d'informatique fondamentale, notion de référence pour celle plus large d'informatique tout court, mais bien plutôt de faire un peu le point de la situation alors qu'on commence à maîtriser mieux la sémantique. Mais j'aimerais par contre qu'il soit en quelque sorte l'ouverture d'un débat, au moins parmi les gens dont on n'entend pas parler mais pour qui l'informatique n'est pas exclusivement le moyen, jupe fendue et talons hauts, d'aller briller sur les trottoirs du pouvoir parisien.**



## VOUZZAVEDIBISAR

Travaux d'Edmond Bianco et dirigés par Edmond Bianco

*Jean - Michel Knippel*

Le numéro 70 de notre périodique revenait sur les éditoriaux écrits au gré du temps et de l'humeur. L'éditorial que vous venez de lire dans ce numéro a été terminé quelques jours ou quelques heures avant le décès d'Edmond Bianco. Je me suis permis de corriger les fautes d'orthographe de plus en plus fréquentes vers les derniers paragraphes. Le reste est sans aucune modification.

Dans le numéro 71, nous avons parcouru les rubriques VOUZZAVEDIBISAR, histoires souvent imagées suivant les périodes. La rubrique continuera. Ce ne sont pas les "VOUZZAVEDIBISAReries" qui manquent pour alimenter ces pages. Les propositions ne manquent pas et je m'en réjouis.

La collection de l'année 2005 se termine par ce numéro centré sur les articles écrits par Edmond Bianco dans le bulletin et les travaux originaux qu'il a faits ou dirigés pendant une quarantaine d'années.

Le récapitulatif est donné par ordre chronologique. Vous trouverez les divers travaux d'Edmond Bianco qui l'ont mené de la faculté des Sciences de Marseille en 1959 à la faculté des Sciences de Paris (1964,1969), puis après Strasbourg (1971), enfin de nouveau à Marseille (1976, 1979)

A ma connaissance, vous avez ici l'essentiel des thèses qu'il a encadrées et dirigées que ce soit des doctorats de 3ème cycle, des doctorats en sciences, des thèses d'état ou habilitations à diriger des recherches, l'essentiel est là. Ceci forme un tout cohérent.

J'y ai rajouté quelques mémoires qui ont été unanimement reconnus pour leur excellence.

Le système universitaire de documentation m'a été d'une aide précieuse pour m'éviter des oublis car Edmond Bianco était discret et ne fréquentait que peu les colloques, conférences et congrès pour y parler ou présenter ses travaux. Par contre, lorsque vous veniez à lui, c'est alors sans compter que la parole fusait et qu'il vous consacrait du temps. Peu importe votre niveau, tout le monde repartait avec des idées à creuser et développer.

Il est certain que j'ai oublié des personnes, mais je pense peu. Vous pouvez me contacter pour rendre cette liste la plus complète possible. Je vous en remercie par avance.

E. BIANCO

Courbure de l'onde de choc attachée

Thèse pour obtenir le grade de Docteur 3<sup>o</sup> cycle

Faculté des Sciences de Marseille. 1959

E. BIANCO et E. LEMAIRE

Algol réduit sur calculateur PB 250

CNRS. Institut Blaise Pascal. Faculté des Sciences de Paris. 1964

E. BIANCO

Étude et formalisation d'une classe de systèmes

Thèse pour obtenir le grade de Docteur d'Etat ès Sciences

Université de Paris (1896-1968). 1969

E. BIANCO, Ph FLAJOLET, P. JAMMES, L. JULIEN, B. ROBINET et R. STUTZMANN

Journées des mathématiques de la compilation

Université de Strasbourg. 1971

E. BIANCO et J. BERSTEL

Journées des mathématiques de la compilation

Université d'Aix-Marseille II. 1973

E. BIANCO

Informatique fondamentale :

de la machine de Turing aux ordinateurs modernes

Cours de C4 et de DEA

Université d'Aix-Marseille II. 1976

P. ISOARDI

Structures de commutation

Thèse pour obtenir le grade de Docteur 3<sup>o</sup> cycle

Université d'Aix-Marseille II. 1978

E. BIANCO

Informatique fondamentale :

de la machine de Turing aux ordinateurs modernes

Éditions Birkhäuser. 1979

J. Ph. LEHMANN

Machines et calculabilité

Thèse pour obtenir le grade de Docteur 3<sup>o</sup> cycle

Université d'Aix-Marseille II. 1984

J. Ph. LEHMANN et P. ISOARDI

Nouvelle structure de mémoires adressables par une fenêtre à déplacement continu

Institut de recherche de mathématique avancée. Strasbourg. 1985

S. EL GHAZI EL HOUSSAINI

Notions fondamentales de la théorie des systèmes informatiques

Thèse pour obtenir le grade de Docteur 3° cycle

Université d'Aix-Marseille II. 1986

A. - L. COURBIS

Les systèmes de Post

Mémoire M.M.I.A.Ge

Universités d'Aix-Marseille II-III. 1987

M. - T. LASKRI

Approche de l'automatisation de thesaurus :

étude de la sémantique adaptée du langage naturel

Thèse pour obtenir le grade de Docteur 3° cycle

Université d'Aix-Marseille II. 1987

P. ISOARDI

Calculateur à mémoire-champ : nouvelle structure de mémoires, processeur associé

Thèse pour obtenir le grade de Docteur d'Etat ès Sciences

Université d'Aix-Marseille II. 1987

J. Ph. LEHMANN

Machines sans changement d'état : aspects théoriques, aspects câblés

Thèse pour obtenir le grade de Docteur d'Etat ès Sciences

Université d'Aix-Marseille II. 1987

N. MESLI

Programmation en procédure formelle d'un compilateur

pour un langage expérimental de recherche et de calcul

C.E.S.U. en informatique fondamentale

Université d'Aix-Marseille II. 1987

CH. BISIÈRE, D. IWANESKO, J. L. MASSAT

Compilateur de machine universelle : MU

Mémoire M.M.I.A.Ge

Universités d'Aix-Marseille II-III. 1989

O. KETTANI

Modèles de calcul sans changement d'état.

Quelques développements et résultats

Thèse pour obtenir le grade de Docteur 3° cycle

Université d'Aix-Marseille II. 1989

E. BIANCO

Essai sur la complexité des systèmes informatiques

Revue internationale de systémique. 1990

Vol.4, N° 2, 1990, pp. 189 à 201

S. HILALA

Système de gestion du parallélisme, contrôlé par système expert

Langage de conduite de processus : L.C.P.

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1991

M. HADJALI

Formalisation des systèmes informatiques ouverts

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1991

P. SANCHEZ

Synthèse de sons musicaux : étude d'une méthode de compilation  
du langage MUSIC V

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1992

E. BOUDIBA

Conception d'un système de gestion de base de données ouvert à  
l'aide du langage BD\_SE

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1993

A. DJABI

Propriétés fondamentales des transferts de l'information dans les  
systèmes universels ouverts

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1993

E. ANDRIAMASINORO

Insertion fractionnée, systèmes et machine universelle

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1994

Y. C. ABOUDI JOELSON

Ordinateur et apprentissage : mémoire souple

Thèse pour obtenir le grade de Docteur en Sciences

Université d'Aix-Marseille II. 1994

J. - M. KNIPPEL

Expression et vérification des thesaurus par les réseaux de Petri.

Cas du répertoire de Kent. Implémentation à l'aide d'une machine universelle

Travaux pour obtenir le diplôme d'Habilitation à diriger des recherches

Université d'Aix-Marseille II. 1994

M. - T. LASKRI

Sémantique du langage naturel à travers un système support de thesaurus

Thèse pour obtenir le grade de Docteur d'Etat en Informatique

Université d'Annaba. 1995